

Programowanie obiektowe

Zadania lab2

Piotr Błaszyński

1 kwietnia 2022

Klasy z poszczególnych zadań zapisywać w oddzielnych plikach, jeżeli w zadaniu jest kilka klas, to można je (przynajmniej na razie) też przechowywać w jednym pliku, jedna po drugiej. Wszystkie klasy pochodne przetestować (niektóre bazowe wg uznania), tworząc kilka obiektów klasy i uruchamiając każdą z dostępnych metod przynajmniej raz.

1. Przygotować klasę bazową `Zwierze` i 2 klasy pochodne: `Pies` i `Kot`, oprócz `init` (podajemy imię) powinna być metoda `"odgłos"`. Stworzyć kilka zwierząt, zebrać je w kolekcji (lista) i wygenerować dla każdego `odgłos`.
2. Przygotować klasę do przechowywania informacji o kontach (imię i nazwisko lub nazwa, numer konta), na podstawie tej klasy utworzyć klasy pochodne opisujące konto firmowe i konto prywatne, konto firmowe powinno mieć dodatkowe metody do płacenia należności (np. `ZUS` i `PIT` - płacenie polega na wyświetleniu pełnej informacji). Ponadto obydwie klasy mają metodę wyświetlającą informację o koncie.
 - a) Utworzyć przynajmniej po trzy nazwane obiekty każdej klasy. Z każdego obiektu wypisać dane podstawowe (można do tego przygotować pomocniczą metodę)
 - b) obiekty przechowujące konta firmowe zapisać na liście, wywołać obydwie metody (płacenia należności) z listy.
 - c) dołożyć metody do zapisu i odczytu z pliku stanu kont i ich numerów (można stosować sposób - jeden plik jeden obiekt) - zapisać obiekty do pliku
3. Do poprzedniego zadania dopisać klasy przechowujące informacje o należnościach (wysokość kwoty i nr konta na który należy wpłacać), proponowana

nazwa klasy bazowej to TaxInfo a pochodne np. ZusInfo i PitInfo. Proszę zwrócić uwagę na kolejność implementacji klas. Skorzystać z nowo utworzonych klas w klasach z poprzedniego zadania, odpytać się klas Info o wartości wyświetlane przy płaceniu należności.

4. Przygotować klasy opisujące konta, tym razem z podziałem na obrotowe i oszczędnościowe,
 - a) obydwie konta mają mieć możliwość wykonywania przelewów pomiędzy kontami (podajemy nr konta i kwotę),
 - b) konto oszczędnościowe może wykonać jeden przelew w określonym miesiącu, dodać też metodę ustawiającą licznik możliwych przelewów na 1 i naliczającą odsetki na kontach oszczędnościowych(koniec miesiąca),
 - c) w init podajemy kwotę początkową,
 - d) zasymulować działanie kilku kont, przelewy pomiędzy kontami, w czasie kilku miesięcy.
 - e) dołożyć metody do zapisu i odczytu z pliku stanu kont i ich numerów (można stosować sposób - jeden plik jeden obiekt - zapisać obiekty do pliku)
5. Wykorzystując klasę Osoba, przygotować klasę Pracownik a do niej dołączyć klasę Wypłata, wykonać przesyłanie Wypłaty do pracownika. W tym zadaniu proszę o samodzielne zaprojektowanie klas.

Dla przypomnienia, przykładowa klasa ręka, dziedzicząca z Organ i korzystanie z niej:

```
class Organ:
    pass
class Reka (Organ):
    def __init__(self , ktora):
        self.ktora=ktora
    def pokaz(self):
        print(self.ktora)

prawa = Reka("PRAWA")
lewa = Reka("LEWA")

prawa.pokaz()
lewa.pokaz()
```

W poniższych przykładach 'w' oznacza zapis do, a 'r' odczyt z pliku. Odczyt pliku:

```
plik = open('nazwa_pliku')
try:
    tekst = plik.read()
finally:
    plik.close()

print(tekst)
```

Zapis do pliku:

```
plik = open('plik.txt', 'w')
try:
    plik.write("tresc")
finally:
    plik.close()
```

Lepszy sposób na odczyt pliku (nie trzeba m.in. pamiętać o close):

```
with open('plik.txt', 'r') as plik:
    tekst = plik.read()
```

Lepszy sposób na zapis do pliku (nie trzeba m.in. pamiętać o close):

```
with open('plik.txt', 'w') as plik:
    plik.write("tresc")
```

Sposób na zapis do pliku kilku linii oddzielonych znakami końca linii (wewnątrz bloku with może być też użyta pętla):

```
with open('plik.txt', 'w') as plik:
    plik.write(str(linia_a)+'\n')
    plik.write(str(linia_b)+'\n')
    plik.write(str(linia_c)+'\n')
```